日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２０００年　７月２４日

出願番号
Application Number: 特願２０００－２２２８９６

出願人
Applicant(s): ソニー株式会社

２００１年　５月２５日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

出証番号　出証特２００１－３０４４７２８

【書類名】　特許願
【整理番号】　00004975
【提出日】　平成12年 7月24日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 17/60
【発明の名称】　情報処理システム及び情報処理方法、並びに記憶媒体
【請求項の数】　13
【発明者】
【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内
【氏名】　末吉　正弘
【特許出願人】
【識別番号】　000002185
【氏名又は名称】　ソニー株式会社
【代表者】　出井　伸之
【代理人】
【識別番号】　100101801
【弁理士】
【氏名又は名称】　山田　英治
【電話番号】　03-5541-7577
【選任した代理人】
【識別番号】　100093241
【弁理士】
【氏名又は名称】　宮田　正昭
【電話番号】　03-5541-7577
【選任した代理人】
【識別番号】　100086531
【弁理士】
【氏名又は名称】　澤田　俊夫

【電話番号】　03-5541-7577
【手数料の表示】
【予納台帳番号】　062721
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1
【物件名】　要約書　1
【包括委任状番号】　9904833
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　情報処理システム及び情報処理方法、並びに記憶媒体

【特許請求の範囲】

【請求項１】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理システムであって、タスク起動時にオペレーティング・システムとタスクの相互認証を実行する起動時認証手段を含むことを特徴とする情報処理システム。

【請求項２】

前記起動時認証手段は、タスクを記述するユーザによって与えられた鍵を用いて相互認証を行うことを特徴とする請求項１に記載の情報処理システム。

【請求項３】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理システムであって、タスクがオペレーティング・システムに対してサービス要求を行うときにオペレーティング・システムとタスクの相互認証を実行するサービス要求時認証手段を含むことを特徴とする情報処理システム。

【請求項４】

前記サービス要求時認証手段は、今回の相互認証に用いた鍵情報で所定のデータを暗号化したデータを次の相互認証に用いる実行用鍵とすることを特徴とする請求項３に記載の情報処理システム。

【請求項５】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理方法であって、タスク起動時にオペレーティング・システムとタスクの相互認証を実行する起動時認証ステップを含むことを特徴とする情報処理方法。

【請求項６】

前記起動時認証ステップでは、タスクを記述するユーザによって与えられた鍵を用いて相互認証を行うことを特徴とする請求項５に記載の情報処理方法。

【請求項７】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理方法であって、タスクがオペレーティング・システムに対してサービス要求を行うときにオペレーティング・システムとタスクの相互認証を実行するサービス要求時認証ステップを含むことを特徴とする情報処理方法。

【請求項８】

前記サービス要求時認証ステップでは、今回の相互認証に用いた鍵情報で所定のデータを暗号化したデータを次の相互認証に用いる実行用鍵とすることを特徴とする請求項７に記載の情報処理方法。

【請求項９】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理方法であって、

オペレーティング・システムが、タスクが持つ相互認証鍵を基にタスクを認証する認証ステップと、

オペレーティング・システムが、タスクのスタック・ポインタを相互認証鍵で暗号化して、タスクに返すステップと、

タスクが、相互認証鍵で暗号化されたスタック・ポインタを復号化して認証するステップと、

を具備することを特徴とする情報処理方法。

【請求項１０】

オペレーティング・システム及びタスクは、認証手続に成功したときには、相互認証鍵で暗号化されたスタック・ポインタを次回以降の認証手続に用いる実行用鍵として保存することを特徴とする請求項８に記載の情報処理方法。

【請求項１１】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理方法であって、

タスクが、実行用鍵を付してオペレーティング・システムに対してサービス要求するステップと、

オペレーティング・システムが、実行用鍵を基にタスクを認証するステップと、

認証に成功したことに応答して、オペレーティング・システムが、依頼されたサービスを実行するとともに、タスクのスタック・ポインタを実行用鍵で暗号化して、次回の実行用鍵を生成するステップと、

オペレーティング・システムが、依頼されたサービスの実行結果とともに次回の実行用鍵をタスクに返すステップと、

を具備することを特徴とする情報処理方法。

【請求項１２】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理をコンピュータ・システム上で実行するように記述されたコンピュータ・ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、前記コンピュータ・ソフトウェアは、

オペレーティング・システムが、タスクが持つ相互認証鍵を基にタスクを認証する認証ステップと、

オペレーティング・システムが、タスクのスタック・ポインタを相互認証鍵で暗号化して、タスクに返すステップと、

タスクが、相互認証鍵で暗号化されたスタック・ポインタを復号化して認証するステップと、

認証手続に成功したときには、オペレーティング・システム及びタスクは相互認証鍵で暗号化されたスタック・ポインタを次回以降の認証手続に用いる実行用鍵として保存するステップと、

を具備することを特徴とする記憶媒体。

【請求項１３】

オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理をコンピュータ・システム上で実行するように記述されたコンピュータ・ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、前記コンピュータ・ソフトウェアは、

タスクが、実行用鍵を付してオペレーティング・システムに対してサービス要求するステップと、

オペレーティング・システムが、実行用鍵を基にタスクを認証するステップと

認証に成功したことに応答して、オペレーティング・システムが、依頼されたサービスを実行するとともに、タスクのスタック・ポインタを実行用鍵で暗号化して、次回の実行用鍵を生成するステップと、

オペレーティング・システムが、依頼されたサービスの実行結果とともに次回の実行用鍵をタスクに返すステップと、

を具備することを特徴とする記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、プロセッサがメモリその他のハードウェア資源を用いてタスクを実行する情報処理システム及び情報処理方法に係り、特に、オペレーティング・システムが提供する環境下で１以上のタスクが動作する構成を備えた情報処理システム及び情報処理方法に関する。

【０００２】

更に詳しくは、本発明は、オペレーティング・システムとオペレーティング・システム上にマルチタスクが存在する構成において、オペレーティング・システムがタスク部分に記述したアプリケーション・プログラムをセキュアに実行する情報処理システム及び情報処理方法に関する。

【０００３】

【従来の技術】

昨今、情報処理や情報通信などのコンピューティング技術が飛躍的に向上し、コンピュータ･システムが広汎に普及してきている。コンピュータ・システムは、一般に、オペレーティング・システム（ＯＳ）の制御下で各種の演算処理を実行する。

【０００４】

最近のオペレーティング・システムは、複数のプログラムを切り替えながら処理し、幾つかの作業を並行して進められるようにする機構、すなわち「マルチテスク」を備えることが一般的となってきている。オペレーティング・システムは

、実際には有限なハードウェア資源を仮想的に多重化して、各プログラムの要求に応じてハードウェア資源を効率よく振り分けるようになっている。

〔０００５〕

例えば、特開２０００－２８２８３号公報には、再生装置とデータ処理装置間で相互認証可能で、再生装置上で再生されるコンテンツ毎にマルチタスク構成を可能とする情報処理方法が開示されている。すなわち、再生装置は、データ処理装置からの識別子ＩＤにより媒体に記録されたデータをデータ処理装置の要求に従って読み出し、中間鍵ＩＮＴ　ＫＥＹを一時記憶することによって複数の暗号化及び認証作業を選択的且つ時分割処理で行い時分割伝送する。また、データ処理装置は、時分割処理によって要求される複数のデータを、再生装置から返される識別子により複数の認証作業及び復号化のためのパラメータを一時記憶することによって選択的且つ時分割処理で行い、複数の時分割処理に対応した暗号化及び認証作業を行う。

〔０００６〕

しかしながら、同公報に記載されている情報処理方法では、相互認証は装置間でのみ行うものであり、タスク起動時やタスクからオペレーティング・システムへサービスを要求する際には、お互いの相互認証を行っていない。

〔０００７〕

カーネル部とタスクが独立して提供される場合、又は、一部のタスクが第３者により製作された場合には、そのタスクが信頼でき且つ正当なものであることを知る手段はない。

〔０００８〕

すなわち、タスク部分に記述したアプリケーション・プログラムをオペレーティング・システムから見てセキュアに実行できない場合がある。

〔０００９〕

〔発明が解決しようとする課題〕

本発明の目的は、オペレーティング・システムが提供する環境下で１以上のタスクが動作する構成を備えた、優れた情報処理システム及び情報処理方法を提供することにある。

【００１０】

本発明の更なる目的は、オペレーティング・システムとオペレーティング・システム上にマルチタスクが存在する構成において、オペレーティング・システムがタスク部分に記述したアプリケーション・プログラムをセキュアに実行することができる、優れた情報処理システム及び情報処理方法を提供することにある。

【００１１】

【課題を解決するための手段】

本発明は、上記課題を参酌してなされたものであり、その第１の側面は、オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理システム又は情報処理方法であって、タスク起動時にオペレーティング・システムとタスクの相互認証を実行する起動時認証手段又はステップを含むことを特徴とする情報処理システム又は情報処理方法である。

【００１２】

ここで、前記起動時認証手段又はステップは、タスクを記述するユーザによって与えられた鍵を用いて相互認証を行うようにしてもよい。

【００１３】

また、本発明の第２の側面は、オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理システム又は情報処理方法であって、タスクがオペレーティング・システムに対してサービス要求を行うときにオペレーティング・システムとタスクの相互認証を実行するサービス要求時認証手段又はステップを含むことを特徴とする情報処理システム又は情報処理方法である。

【００１４】

ここで、前記サービス要求時認証手段は、今回の相互認証に用いた鍵情報で所定のデータを暗号化したデータを次の相互認証に用いる実行用鍵とするようにしてもよい。例えば、暗号化されるデータとしてスタック・ポインタを利用する場合、スタック・ポインタの移動を伴う限り、タスクがオペレーティング・システムへサービスを要求するための鍵が毎回変更されるので、セキュリティが維持される。

【００１５】

また、本発明の第３の側面は、オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理方法であって、

オペレーティング・システムが、タスクが持つ相互認証鍵を基にタスクを認証する認証ステップと、

オペレーティング・システムが、タスクのスタック・ポインタを相互認証鍵で暗号化して、タスクに返すステップと、

タスクが、相互認証鍵で暗号化されたスタック・ポインタを復号化して認証するステップと、

を具備することを特徴とする情報処理方法である。

【００１６】

ここで、オペレーティング・システム及びタスクは、認証手続に成功したときには、相互認証鍵で暗号化されたスタック・ポインタを次回以降の認証手続に用いる実行用鍵として保存するようにしてもよい。

【００１７】

また、本発明の第４の側面は、オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理方法であって、

タスクが、実行用鍵を付してオペレーティング・システムに対してサービス要求するステップと、

オペレーティング・システムが、実行用鍵を基にタスクを認証するステップと、

認証に成功したことに応答して、オペレーティング・システムが、依頼されたサービスを実行するとともに、タスクのスタック・ポインタを実行用鍵で暗号化して、次回の実行用鍵を生成するステップと、

オペレーティング・システムが、依頼されたサービスの実行結果とともに次回の実行用鍵をタスクに返すステップと、

を具備することを特徴とする情報処理方法である。

【００１８】

また、本発明の第５の側面は、オペレーティング・システムが提供する実行環

境下で１以上のタスクが実行される情報処理をコンピュータ・システム上で実行するように記述されたコンピュータ・ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、前記コンピュータ・ソフトウェアは、

オペレーティング・システムが、タスクが持つ相互認証鍵を基にタスクを認証する認証ステップと、

オペレーティング・システムが、タスクのスタック・ポインタを相互認証鍵で暗号化して、タスクに返すステップと、

タスクが、相互認証鍵で暗号化されたスタック・ポインタを復号化して認証するステップと、

認証手続に成功したときには、オペレーティング・システム及びタスクは相互認証鍵で暗号化されたスタック・ポインタを次回以降の認証手続に用いる実行用鍵として保存するステップと、

を具備することを特徴とする記憶媒体である。

【００１９】

また、本発明の第６の側面は、オペレーティング・システムが提供する実行環境下で１以上のタスクが実行される情報処理をコンピュータ・システム上で実行するように記述されたコンピュータ・ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、前記コンピュータ・ソフトウェアは、

タスクが、実行用鍵を付してオペレーティング・システムに対してサービス要求するステップと、

オペレーティング・システムが、実行用鍵を基にタスクを認証するステップと、

認証に成功したことに応答して、オペレーティング・システムが、依頼されたサービスを実行するとともに、タスクのスタック・ポインタを実行用鍵で暗号化して、次回の実行用鍵を生成するステップと、

オペレーティング・システムが、依頼されたサービスの実行結果とともに次回の実行用鍵をタスクに返すステップと、

を具備することを特徴とする記憶媒体である。

【００２０】

本発明の第５及び第６の各側面に係る記憶媒体は、例えば、様々なプログラム・コードを実行可能な汎用性のコンピュータ・システムに対して、コンピュータ・ソフトウェアをコンピュータ可読な形式で物理的に提供する媒体である。このような媒体は、例えば、ＣＤ（Compact Disc）やＦＤ（Floppy Disc）、ＭＯ（Magneto-Optical disc）などの着脱自在で可搬性の記憶媒体である。あるいは、ネットワーク（ネットワークは無線、有線の区別を問わない）などの伝送媒体などを経由してコンピュータ・ソフトウェアを特定のコンピュータ・システムにコンピュータ可読形式で提供することも技術的に可能である。

【００２１】

このような記憶媒体は、コンピュータ・システム上で所定のコンピュータ・ソフトウェアの機能を実現するための、コンピュータ・ソフトウェアと記憶媒体との構造上又は機能上の協働的関係を定義したものである。換言すれば、本発明の第５及び第６の各側面に係る記憶媒体を介して所定のコンピュータ・ソフトウェアをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発明の第３及び第４の各側面に係る情報処理方法と同様の作用効果を得ることができる。

【００２２】

【作用】

本発明に係る情報処理システムは、オペレーティング・システムとオペレーティング・システム上で実行されるタスクで構成され、タスク起動時にタスクがオペレーティング・システム側と相互認証を行い、タスクの正当性を判定する機構を備えている。

【００２３】

例えば、第３者が製作したタスクの起動時点又はタスクがサービスをオペレーティング・システムに要求する時点での認証や検証が可能である。

【００２４】

また、タスクがオペレーティング・システムに対してサービスを要求するときに、オペレーティング・システムはタスクが持つ鍵を評価して、オペレーティング・システム自身が持つ鍵と同一である場合のみサービスの実行を許可するよう

にすることで、セキュリティを維持することができる。

【００２５】

本発明によれば、オペレーティング・システム側は、タスク毎にスタック・ポインタを鍵で暗号化したデータを次回の鍵（ＩＤ）として使用するようにしている。したがって、スタック・ポインタの移動を伴う限り、タスクがオペレーティング・システムへサービスを要求する鍵が毎回変更されるので、セキュリティが維持される。

【００２６】

以上を総括すれば、本発明に係る情報処理システム及び情報処理方法によれば、タスク部分に記述したアプリケーション・プログラムを、オペレーティング・システム側から見てセキュアに実行することができる訳である。

【００２７】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００２８】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施例を詳解する。

【００２９】

本発明を実現する上で、アプリケーションをタスクとして記述するユーザ（以下では、単に「ユーザ」とする）は、下表に示すように、タスク番号（又は、タスクＩＤ）毎に相互認証用の鍵を与えているものとする。

【００３０】

【表１】

タスクＩＤ	タスク・イニシャル・キー
1	Ｘ１，Ｘ２，…，Ｘ８
2	Ｙ１，Ｙ２，…，Ｙ８
⋮	⋮

【００３１】

一方、オペレーティング・システム側は、タスクＩＤと鍵の管理テーブルを、ユーザが読み書きできない状態で内部に保持する。

【００３２】

図１には、鍵用管理テーブルの構成と該テーブルが格納される領域について模式的に図解している。

【００３３】

タスクの鍵管理テーブルでは、各タスクＩＤ毎に、相互認証用鍵（タスク・イニシャル・キー）と実行用鍵（タスク・ワーキング・キー）が設定される。

【００３４】

また、タスク毎に鍵格納領域と、現在のスタック・ポインタを知る関数”GesCurrentSP()”をユーザ側で用意する。

【００３５】

図２には、関数とタスク・コントロール・ブロックの作用について模式的に図解している。

【００３６】

オペレーティング・システム側では、タスク分だけタスク・コントロール・ブロック（ＴＣＢ）が用意されている。タスク・コントロール・ブロック中には、オペレーティング・システムが最後にサービスを受けたときのスタック・ポインタが格納されている。また、オペレーティング・システムは、タスクＩＤを返す

関数"TaskID()"を持っている。

〔００３７〕

タスクがタスクＩＤを取得するためには、オペレーティング・システムに対するサービス要求"TaskID()"を用いればよい。また、タスク側でスタック・ポインタを取得するには、ユーザ側で用意した関数"GetCurrentSP()"を用いればよい。

〔００３８〕

オペレーティング・システムのスケジューラ起動に伴ったタスク切替時に、そのタスクが実行されていた時点でのスタック・ポインタをタスク・コントロール・ブロック中の所定フィールドに毎回保持するようになっている。

〔００３９〕

次いで、タスクとオペレーティング・システムが相互認証を行う処理手順について、図３に示したフローチャートを参照しながら説明する。

〔００４０〕

まず、タスクは、関数"TaskID()"を用いて、オペレーティング・システム側から自身のタスクＩＤを取得する（ステップＳ１）。そして、タスクは、タスクＩＤと相互認証用鍵すなわちタスク・イニシャル・キーを、オペレーティングシステム側に渡す（ステップＳ２）。タスクＩＤや認証鍵の受け渡しは、例えば「スーパーバイザ・コール」によって行われる。

〔００４１〕

オペレーティング・システムは、受け取ったタスク・イニシャル・キーとタスク鍵管理テーブル側で管理しているタスク・イニシャル・キーとを比較する（ステップＳ３及びＳ４）。両者が一致しなければ認証に失敗し、本認証処理ルーチン全体を終了する。

〔００４２〕

一方、鍵が一致する場合には、オペレーティング・システム側における認証が成功したものとみなす。この場合、オペレーティング・システムは、タスクのスタック・ポインタをタスク・イニシャル・キーで暗号化して（ステップＳ５）、この暗号化されたデータ"$Enc_{TIKey}(TaskSP)$"をタスク側に返す（ステップＳ６）。

【００４３】

タスク側は、受け取った暗号化データを、自身のタスク・イニシャル・キーで復号化するとともに（ステップＳ７）、これを関数"GetCurrentSP()"を用いて得られたスタック・ポインタの値と比較する（ステップＳ８）。両者が一致しなければ認証に失敗し、本認証処理ルーチン全体を終了する。

【００４４】

一方、両データが一致する場合には、タスク側の認証が成功したものとみなして、暗号化されたデータ"Enc_{TIKey}(TaskSP)"をタスク鍵領域に保管する（ステップＳ９）。

【００４５】

さらに、オペレーティング・システム側も、タスク・コントロール・ブロック中のスタック・ポインタを暗号化したデータをタスク・イニシャル・キーで暗号化したデータをタスク鍵管理テーブルの実行用鍵エリア中にタスク・ワーキング・キーとして保管する（ステップＳ１０）。

【００４６】

次いで、タスク実行時の認証処理の流れについて、図４を参照しながら説明する。

【００４７】

タスクは、オペレーティング・システムに対するサービスの依頼を、タスク鍵領域の内容すなわちタスク・ワーキング・キーとともに渡す（ステップＳ１１）。このサービスの依頼は、例えば、周辺機器への要求、メモリ・プールの管理、時間管理、他のタスクへのメッセージ送信・受信など、オペレーティング・システムが提供する機能である。また、サービス依頼は、例えばスーパーバイザ・コールによって行われる。

【００４８】

これに対し、オペレーティング・システム側では、スーパーバイザ・コールとタスク・ワーキング・キーとを受け取って（ステップＳ１２）、鍵管理テーブルにおいて保管されている対応するタスクＩＤのそれと比較する（ステップＳ１３及びＳ１４）。両者が一致しなければ認証に失敗し、本認証処理ルーチン全体を

終了する。

【００４９】

一方、鍵が一致する場合には、オペレーティング・システムはサービス依頼元であるタスクに実行権があると判断する。

【００５０】

この場合、オペレーティング・システムは、タスクのスタック・ポインタをタスク・ワーキング・キーで暗号化したデータを、次のタスク・ワーキング・キーとして生成して（ステップＳ１５）、タスク鍵管理テーブル中の対応するタスク・ワーキング・キー・フィールドに書き込む。

【００５１】

そして、オペレーティング・システムは、依頼されたサービスを実行して、その実行結果とともに次回のタスク・ワーキング・キーとしてタスク側に返す（ステップＳ１６）。

【００５２】

タスク側は、サービスの完了と、受け取ったタスク・ワーキング・キーを、タスク鍵領域に格納して、次回の実行に備える。

【００５３】

［追補］

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【００５４】

【発明の効果】

以上詳記したように、本発明によれば、オペレーティング・システムが提供する環境上で１以上のタスクが動作する構成を備えた、優れた情報処理システム及び情報処理方法を提供することができる。

【００５５】

また、本発明によれば、オペレーティング・システムとオペレーティング・システム上にマルチタスクが存在する構成において、オペレーティング・システムがタスク部分に記述したアプリケーション・プログラムをセキュアに実行することができる、優れた情報処理システム及び情報処理方法を提供することができる。

【００５６】

本発明に係る情報処理システムは、オペレーティング・システムとオペレーティング・システム上で実行されるタスクで構成され、タスク起動時にタスクがオペレーティング・システム側と相互認証を行い、タスクの正当性を判定する機構を備えている。例えば、第３者が製作したタスクを、その起動時点並びにタスクがサービスをオペレーティング・システムに要求する時点において、認証や検証を行うことが可能である。

【００５７】

また、タスクがオペレーティング・システムに対してサービスを要求するときに、オペレーティング・システムはタスクが持つ鍵を評価して、オペレーティング・システム自身が持つ鍵と同一である場合のみサービスの実行を許可するようにすることで、セキュリティを維持することができる。

【００５８】

本発明によれば、オペレーティング・システム側は、タスク毎にスタック・ポインタを鍵で暗号化したデータを次回の鍵（ＩＤ）として使用するようにしている。したがって、スタック・ポインタの移動を伴う限り、タスクがオペレーティング・システムへサービスを要求するための鍵が毎回変更されるので、セキュリティが維持される。

【００５９】

したがって、タスク部分に記述したアプリケーション・プログラムを、オペレーティング・システム側から見てセキュアに実行することができる。

【図面の簡単な説明】

【図１】

鍵用管理テーブルの構成と該テーブルが格納される領域について模式的に示し

た図である。

【図２】

関数とタスク・コントロール・ブロックを説明するための図である。

【図３】

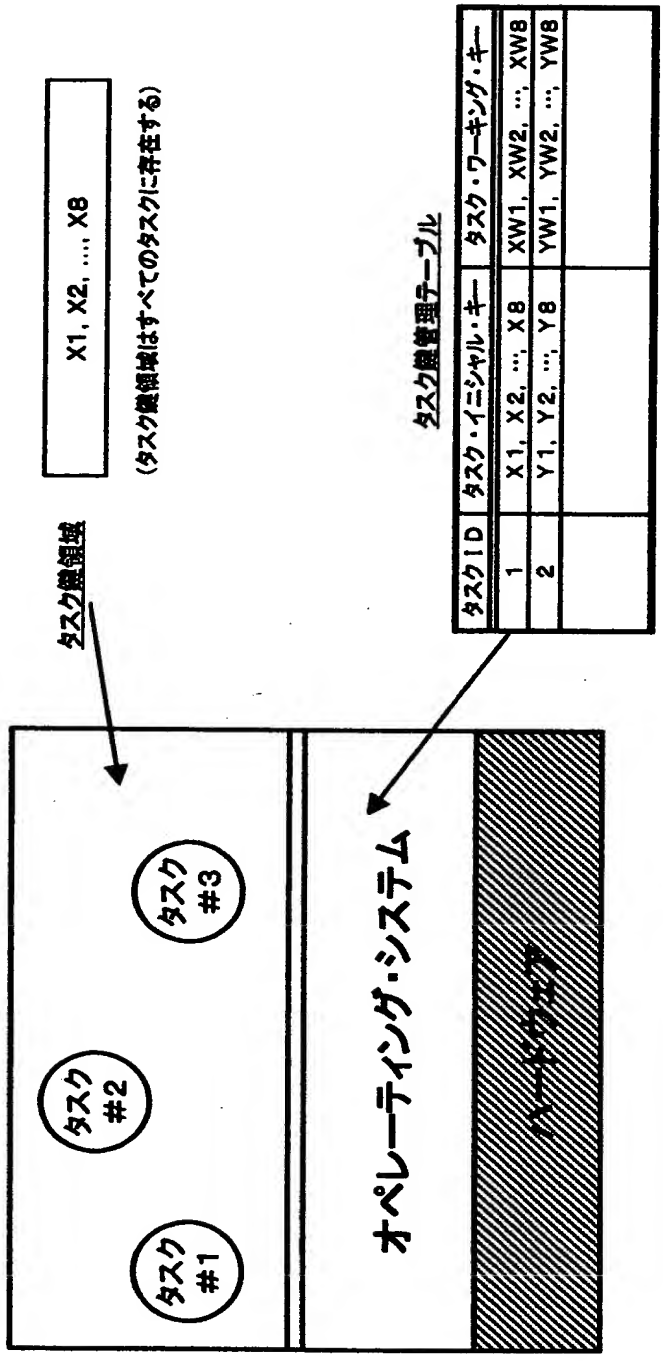
タスクとオペレーティング・システムが相互認証を行う処理手順を示したフローチャートである。

【図４】

サービス認証時の認証処理を行う手順を示したフローチャートである。

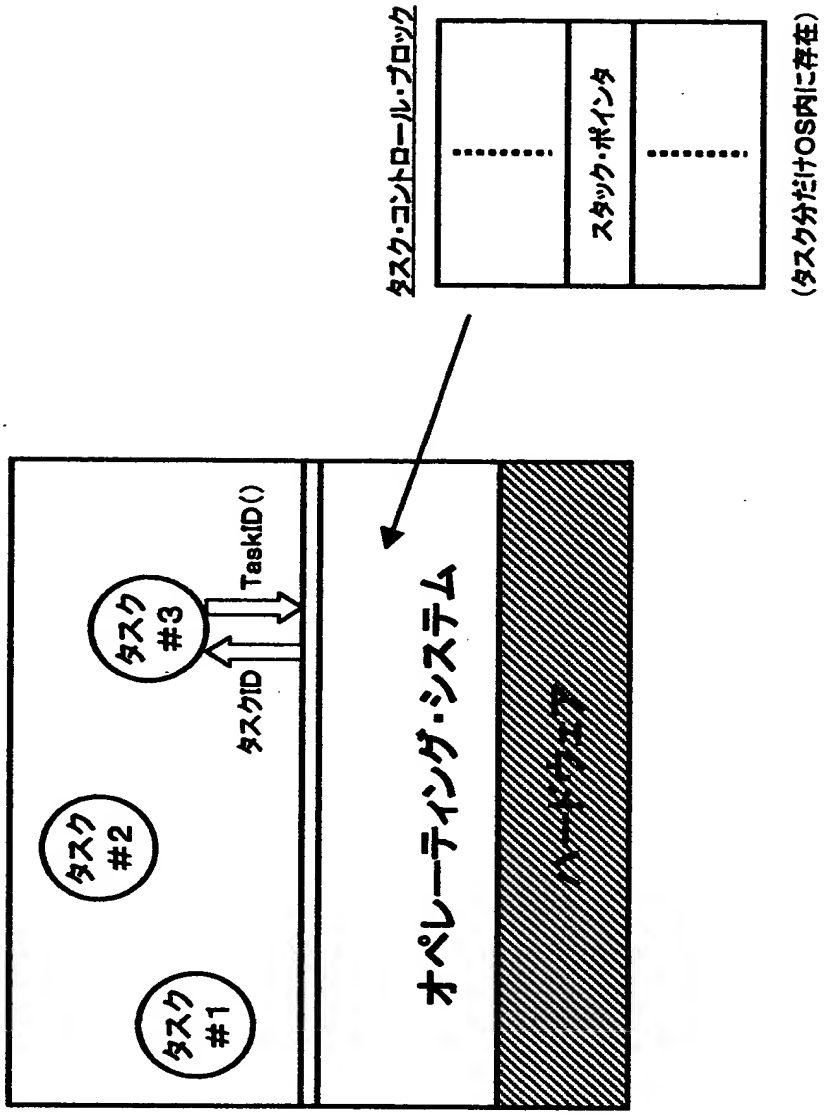
【書類名】　　図面

【図１】

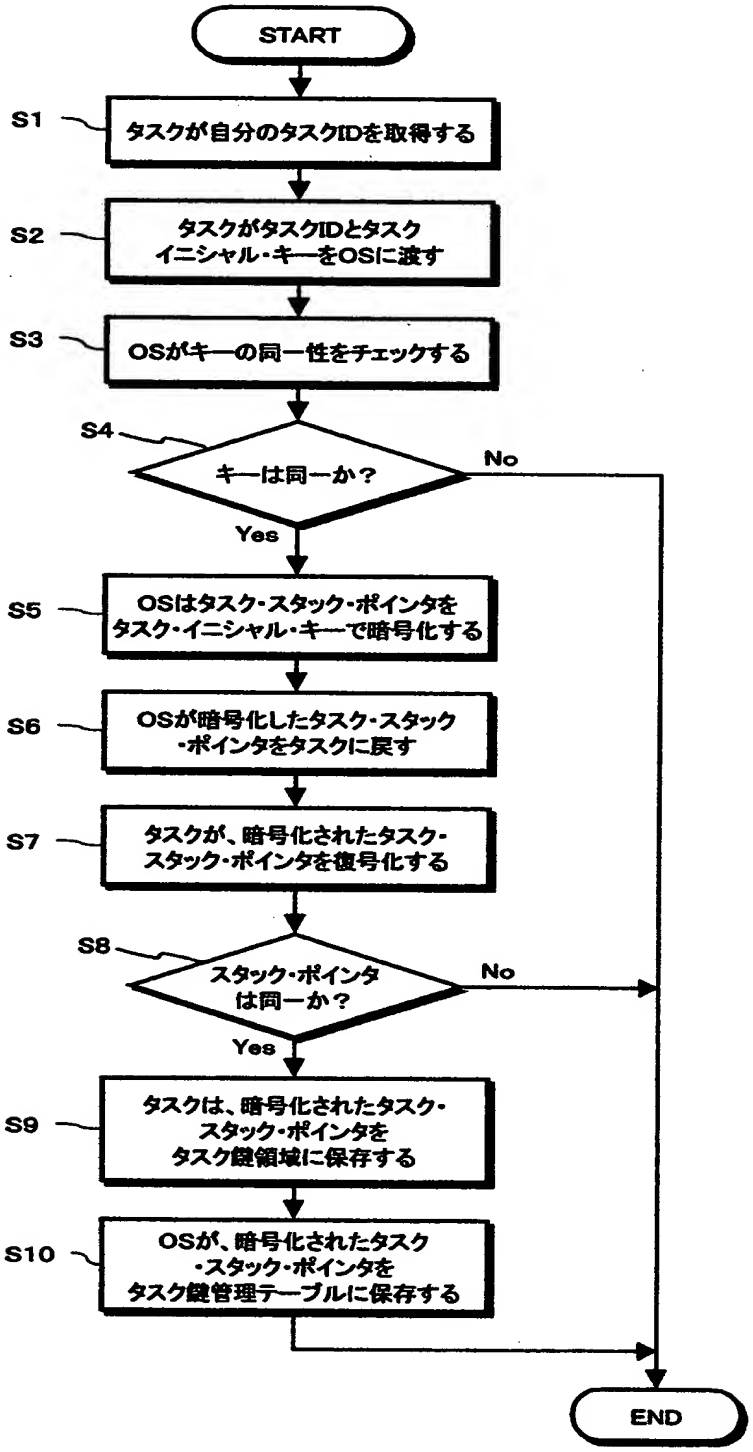


出証特２００１－３０４４７２８

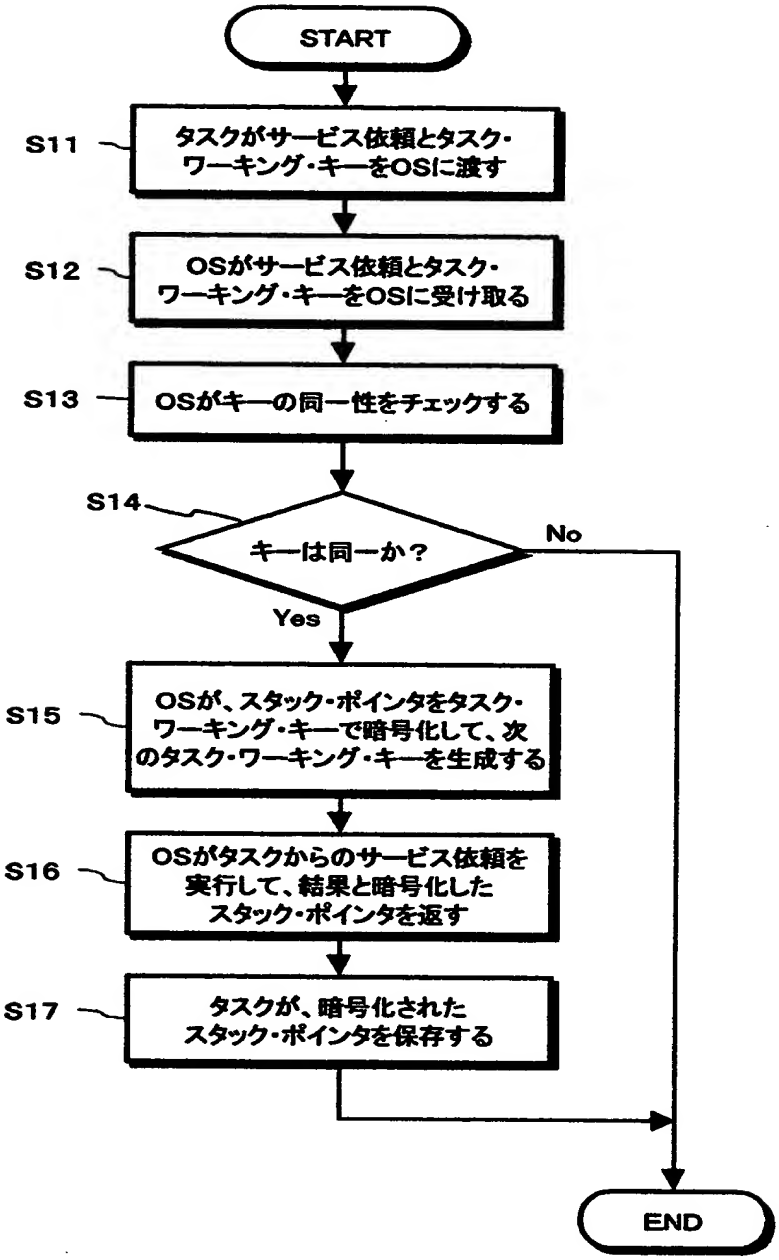
【図２】


タスク・コントロール・ブロック
スタック・ポインタ
(タスク分だけOS内に存在)
TaskID()
タスクID
タスク #3
タスク #2
タスク #1
オペレーティング・システム
ハードウェア

【図３】


START
S1
タスクが自分のタスクIDを取得する
S2
タスクがタスクIDとタスク
イニシャル・キーをOSに渡す
S3
OSがキーの同一性をチェックする
S4
キーは同一か？
No
Yes
S5
OSはタスク・スタック・ポインタを
タスク・イニシャル・キーで暗号化する
S6
OSが暗号化したタスク・スタック
・ポインタをタスクに戻す
S7
タスクが、暗号化されたタスク・
スタック・ポインタを復号化する
S8
スタック・ポインタ
は同一か？
No
Yes
S9
タスクは、暗号化されたタスク・
スタック・ポインタを
タスク鍵領域に保存する
S10
OSが、暗号化されたタスク
・スタック・ポインタを
タスク鍵管理テーブルに保存する
END

【図４】


START
S11
タスクがサービス依頼とタスク・ワーキング・キーをOSに渡す
S12
OSがサービス依頼とタスク・ワーキング・キーをOSに受け取る
S13
OSがキーの同一性をチェックする
S14
キーは同一か？
No
Yes
S15
OSが、スタック・ポインタをタスク・ワーキング・キーで暗号化して、次のタスク・ワーキング・キーを生成する
S16
OSがタスクからのサービス依頼を実行して、結果と暗号化したスタック・ポインタを返す
S17
タスクが、暗号化されたスタック・ポインタを保存する
END

【書類名】　　要約書

【要約】

【課題】　オペレーティング・システムから見て、タスク部分に記述したアプリケーション・プログラムをセキュアに実行可能にする。

【解決手段】　オペレーティング・システムとオペレーティング・システム上で実行されるタスクで構成され、タスク起動時にタスクがオペレーティング・システム側と相互認証を行い、タスクの正当性を判定する機構を備えている。タスクがオペレーティング・システムに対してサービスを要求するときに、オペレーティング・システムはタスクが持つ鍵を評価して、オペレーティング・システム自身が持つ鍵と同一である場合のみサービスの実行を許可する。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００２１８５〕

１．変更年月日　　１９９０年　８月３０日
［変更理由］　新規登録
住　所　　東京都品川区北品川６丁目７番３５号
氏　名　　ソニー株式会社